

# Application of entropy compression in pattern avoidance

Pascal Ochem, Alexandre Pinlou<sup>1</sup>

*LIRMM - Univ. Montpellier 2, CNRS - 161 rue Ada, 34095 Montpellier Cedex 5,  
France*

---

## Abstract

In combinatorics on words, a word  $w$  over an alphabet  $\Sigma$  is said to avoid a pattern  $p$  over an alphabet  $\Delta$  if there is no factor  $f$  of  $w$  such that  $f = h(p)$  where  $h : \Delta^* \rightarrow \Sigma^*$  is a non-erasing morphism. A pattern  $p$  is said to be  $k$ -avoidable if there exists an infinite word over a  $k$ -letter alphabet that avoids  $p$ . We give a positive answer to Problem 3.3.2 in Lothaire's book "Algebraic combinatorics on words", that is, every pattern with  $k$  variables of length at least  $2^k$  (resp.  $3 \times 2^{k-1}$ ) is 3-avoidable (resp. 2-avoidable). This improves previous bounds due to Bell and Goh, and Rampersad.

*Key words:* Word; Pattern avoidance.

---

## 1 Introduction

A pattern  $p$  is a non-empty word over an alphabet  $\Delta = \{A, B, C, \dots\}$  of capital letters called *variables*. A word  $x$  over  $\Sigma$  is an instance of  $p$  if there exists a non-erasing morphism  $h : \Delta^* \rightarrow \Sigma^*$  such that  $h(p) = x$ . The avoidability index  $\lambda(p)$  of a pattern  $p$  is the size of the smallest alphabet  $\Sigma$  such that there exists an infinite word  $w$  over  $\Sigma$  containing no instance of  $p$  as a factor. Bean, Ehrenfeucht, and McNulty [1] and Zimin [15] characterized unavoidable patterns, i.e., such that  $\lambda(p) = \infty$ . We say that a pattern  $p$  is  $t$ -avoidable if  $\lambda(p) \leq t$ . For more informations on pattern avoidability, we refer to Chapter 3 of Lothaire's book [7].

In this paper, we consider upper bounds on the avoidability index of long enough patterns with  $k$  variables. Bell and Goh [2] and Rampersad [11] used

---

*Email addresses:* `Pascal.Ochem@lirmm.fr` (Pascal Ochem),  
`Alexandre.Pinlou@lirmm.fr` (Alexandre Pinlou).  
*URLs:* `http://www.lirmm.fr/~ochem` (Pascal Ochem),  
`http://www.lirmm.fr/~pinlou` (Alexandre Pinlou).

<sup>1</sup> Second affiliation: Département de Mathématiques et Informatique Appliquées, Université Paul-Valéry, Montpellier 3, Route de Mende, 34199 Montpellier Cedex 5, France.

a method based on power series and obtained the following bounds:

**Theorem 1** ([2,11]) *Let  $p$  be a pattern with exactly  $k$  variables.*

- (a) *If  $p$  has length at least  $2^k$  then  $\lambda(p) \leq 4$ . [2]*
- (b) *If  $p$  has length at least  $3^k$  then  $\lambda(p) \leq 3$ . [11]*
- (c) *If  $p$  has length at least  $4^k$  then  $\lambda(p) = 2$ . [11]*

Our main result improves these bounds:

**Theorem 2** *Let  $p$  be a pattern with exactly  $k$  variables.*

- (a) *If  $p$  has length at least  $2^k$  then  $\lambda(p) \leq 3$ .*
- (b) *If  $p$  has length at least  $3 \times 2^{k-1}$  then  $\lambda(p) = 2$ .*

Theorem 2 gives a positive answer to Problem 3.3.2 of Lothaire's book [7]. The bound  $2^k$  in Theorem 2.(a) is tight in the sense that for every  $k \geq 1$ , the pattern  $p_k$  with  $k$  variables in the family  $\{A, ABA, ABACABA, ABACABADABACABA, \dots\}$  has length  $2^k - 1$  and is unavoidable. Similarly, the bound  $3 \times 2^{k-1}$  in Theorem 2.(b) is tight in the sense that for every  $k \geq 1$ , the pattern with  $k$  variables in the family  $\{AA, AABAA, AABAACAABAA, AABAACAABAADAABAACAABAA, \dots\}$  has length  $3 \times 2^{k-1} - 1$  and is not 2-avoidable. Hence, this shows that the upper bound 3 of Theorem 2.(a) is best possible.

The avoidability index of every pattern with at most 3 variables is known, thanks to various results in the literature. In particular, Theorem 2 is proved for  $k \leq 3$ :

- For  $k = 1$ , the famous results of Thue [13,14] give  $\lambda(AA) = 3$  and  $\lambda(AAA) = 2$ .
- For  $k = 2$ , every binary pattern of length at least 4 contains a square, and is thus 3-avoidable. Moreover, Roth [12] proved that every binary pattern of length at least 6 is 2-avoidable.
- For  $k = 3$ , Cassaigne [4] began and the first author [9] finished the determination of the avoidability index of every pattern with at most 3 variables. Every ternary pattern of length at least 8 is 3-avoidable and every binary pattern of length at least 12 is 2-avoidable.

So, there remains to prove the cases  $k \geq 4$ .

Section 2 is devoted to some preliminary results. We prove Theorem 2.(a) in Section 3 as a corollary of a result of Bell and Goh [2]. In Section 4, we prove Theorem 2.(b) using the so-called *entropy compression method*.

## 2 Preliminary results

Let  $p$  be a pattern over  $\Delta = \{A, B, C, \dots\}$ . An *occurrence*  $y$  of  $p$  is an assignment of a non-empty words over  $\Sigma$  to every variable of  $p$  that form a factor. Note that two distinct occurrences of  $p$  may form the same factor. For example, if  $p = ABA$  then the occurrence  $y = (A = 00; B = 1)$  of  $p$  forms the factor

00100; on the other hand,  $y' = (A = 0; B = 010)$  is a distinct occurrence of  $p$  which forms the same factor 00100.

A pattern  $p$  is *doubled* if every variable of  $p$  appears at least twice in  $p$ .

A pattern  $p$  is *balanced* if it is doubled and every variable of  $p$  appears both in the prefix and the suffix of length  $\lfloor \frac{|p|}{2} \rfloor$  of  $p$ . Note that if the pattern has odd length, then the variable  $X$  that appears in the middle of  $p$  (i.e. in position  $\lfloor \frac{|p|}{2} \rfloor + 1$ ) must appear also in the prefix and in the suffix in order to make  $p$  balanced.

**Claim 1** *For every integer  $f \geq 2$ , every pattern with at most  $k$  variables and length at least  $f \times 2^{k-1}$  contains a balanced pattern  $p'$  with at most  $k' \geq 1$  variables and length at least  $f \times 2^{k'-1}$  as a factor.*

**Proof.** We prove this claim by induction on  $k$ . If  $k = 1$ , then  $p$  has size at least  $f \geq 2$  and is clearly balanced. Suppose this is true for some  $k = n$ , i.e.  $p$  with  $n$  variables and length at least  $f \times 2^{n-1}$  contains a balanced pattern  $p'$  as a factor with at most  $k'$  variables and length at least  $f \times 2^{k'-1}$ . Let  $k = n + 1$  and let  $p_1$  (resp.  $p_2$ ) be the prefix (resp. the suffix) of  $p$  of size  $\lfloor \frac{|p|}{2} \rfloor$ . If  $p$  is not balanced, then there exists a variable  $X$  in  $p$  that does not occur in  $p_i$  for some  $i \in \{1, 2\}$ . Thus,  $p_i$  has at most  $k - 1 = n$  variables and length at least  $f \times 2^{n-1}$ . Therefore, by induction hypothesis,  $p$  contains a balanced pattern with at most  $k'$  variables and length at least  $f \times 2^{k'-1}$  as a factor.  $\square$

In the following, we will only use the fact that the pattern  $p'$  in Claim 1 is doubled instead of balanced.

### 3 3-avoidable long patterns

We prove Theorem 2.(a) as a corollary of the following result of Bell and Goh [2]:

**Lemma 3 ([2])** *Every doubled pattern with at least 6 variables is 3-avoidable.*

**Proof of Theorem 2.(a).** We want to prove that every pattern with exactly  $k$  variables and length at least  $2^k$  is 3-avoidable, or equivalently, that every pattern with at most  $k$  variables and length at least  $2^k$  is 3-avoidable. By Claim 1, every such pattern contains a doubled pattern  $p'$  as a factor with at most  $k' \geq 1$  variables and length at least  $2^{k'}$ . So there remains to show that every doubled pattern with at most  $k$  variables and length at least  $2^k$  is 3-avoidable. As discussed in the introduction, the case of patterns with at most 3 variables has been settled. Now, it is sufficient to prove that doubled patterns of length at least  $2^4 = 16$  are 3-avoidable.

Suppose that  $p_1$  is a doubled pattern containing a variable  $X$  that appears at least 4 times. Replace 2 occurrences of  $X$  with a new variable to obtain a pattern  $p_2$ . Example: We replace the first and third occurrence of  $B$  in  $p_1 = ABBCDBCABDDCB$  by a new variable  $E$  to obtain  $p_2 = AEBCDECABDDCB$ . Then  $p_2$  is a doubled pattern such that  $|p_1| = |p_2|$  and  $\lambda(p_1) \leq \lambda(p_2)$ , since every occurrence of  $p_1$  is also an occurrence of  $p_2$ .

Given a doubled pattern  $p$  of length at least 16, we make such replacements as long as we can. We thus obtain a doubled pattern  $p'$  of length at least 16 such that  $\lambda(p) \leq \lambda(p')$ . Moreover, every variable in  $p'$  appears either 2 or 3 times and therefore  $p'$  contains at least  $\lceil 16/3 \rceil = 6$  variables. So  $p'$  is 3-avoidable by Lemma 3. Thus  $p$  is 3-avoidable, which finishes the proof.  $\square$

## 4 2-avoidable long patterns

We want to prove that every pattern with exactly  $k$  variables and length at least  $3 \times 2^{k-1}$  is 2-avoidable, or equivalently, that every pattern with at most  $k$  variables and length at least  $3 \times 2^{k-1}$  is 2-avoidable. By Claim 1, every such pattern contains a doubled pattern  $p'$  as a factor with at most  $k' \geq 1$  variables and length at least  $3 \times 2^{k'-1}$ . So there remains to show that every doubled pattern with at most  $k$  variables and length at least  $3 \times 2^{k-1}$  is 2-avoidable.

As discussed in the introduction, the case of patterns with at most 3 variables has been settled. Now, it is sufficient to prove Theorem 2.(b) for doubled patterns and  $k \geq 4$ .

Let  $\Sigma = \{0, 1\}$  be the alphabet. For the remaining of this section, let  $k \geq 4$  and  $q(k) = 3 \times 2^{k-1}$ .

Suppose by contradiction that there exists a doubled pattern  $p$  on  $k$  variables and length at least  $q(k)$  that is not 2-avoidable. Then there exists an integer  $n$  such that any word  $w \in \Sigma^n$  contains  $p$ . We put an arbitrary order on the  $k$  variables of  $p$  and call  $A_j$  the  $j$ -th variable of  $p$ .

### 4.1 The algorithm AVOIDPATTERN

Let  $V = \{0, 1\}^t$  be a vector of length  $t$ . The following algorithm takes the vector  $V$  as input and returns a word  $w$  avoiding  $p$  and a data structure  $R$  that is called a *record* in the remaining of the paper.

---

#### Algorithm 1: AVOIDPATTERN

---

**Input** :  $V$ .

**Output**:  $w$  (a word avoiding  $p$ ) and  $R$  (a data structure).

---

```

1  $w \leftarrow \varepsilon$ 
2  $R \leftarrow \emptyset$ 
3 for  $i \leftarrow 1$  to  $t$  do
4   Append  $V[i]$  (the  $i$ -th letter of  $V$ ) to  $w$ 
5   Encode in  $R$  that a letter was appended to  $w$ 
6   if  $w$  contains an occurrence of  $p$  of length  $\ell$  then
7     Encode in  $R$  the occurrence of  $p$ 
8     Erase the suffix of length  $\ell$  of  $w$ 
9 return  $R, w$ 
```

---

The way we encode information in  $R$  at lines 5 and 7 will be explained in Subsection 4.2.

In the algorithm AVOIDPATTERN, let  $w_i$  be the word  $w$  after  $i$  steps. Clearly,  $w_i$  avoids  $p$  at each step. By contradiction hypothesis, the resulting word  $w$  of the algorithm (that is  $w_t$ ) has length less than  $n$ . We will prove that each output of the algorithm allows to determine the input. Then we obtain a contradiction by showing that the number of possible outputs is strictly smaller than the number of possible inputs when  $t$  is chosen large enough compared to  $n$ . This implies that every pattern  $p$  with at most  $k$  variables and length at least  $q(k)$  is 2-avoidable.

To analyze the algorithm, we borrow ideas from graph coloring problems [5,6]. These results are based on the Moser-Tardos [8] entropy-compression method which is an algorithmic proof of the Lovász Local Lemma.

#### 4.2 The record $R$

An important part of the algorithm is to keep the record  $R$  of each step of the algorithm. Let  $R_i$  be the record after  $i$  steps of the algorithm AVOIDPATTERN. On one hand, given  $V$  as input of the algorithm, this produces a pair  $(R_t, w_t)$ . On the other hand, given a pair  $(R_t, w_t)$ , we will show in Lemma 4 that we can recover the entire input vector  $V$ . So, each input vector  $V$  produces a distinct pair  $(R_t, w_t)$ .

Let  $\mathcal{V}$  be the set of input vectors  $V$  of size  $t$ , let  $\mathcal{R}$  be the set of records  $R$  produced by the algorithm AVOIDPATTERN and let  $\mathcal{O}$  be the set of different outputs  $(R_t, w_t)$ . After the execution of the algorithm ( $t$  steps),  $w_t$  avoids  $p$  by definition and therefore  $|w_t| < n$  by contradiction hypothesis. Hence, the number of possible final words  $w_t$  is independent from  $t$  (it is at most  $2^n$ ). We then clearly have  $|\mathcal{O}| \leq 2^n \times |\mathcal{R}|$ . We will prove that  $|\mathcal{V}| \leq |\mathcal{O}|$  and that  $|\mathcal{R}| = o(2^t)$  to obtain the contradiction  $2^t = |\mathcal{V}| \leq |\mathcal{O}| \leq 2^n \times |\mathcal{R}| = o(2^t)$ .

The record  $R$  is a triplet  $R = (D, L, X)$  where  $D$  is a binary word (each element is 0 or 1),  $L$  is a vector of  $(k-1)$ -sets of non-zero integers and  $X$  is a vector of binary words. At the beginning,  $D$ ,  $L$  and  $X$  are empty. At step  $i$  of the algorithm, we append  $V[i]$  to  $w_{i-1}$  to get  $w'_i$ .

If  $w'_i$  contains no occurrence of  $p$ , then we append 0 to  $D$  to get  $R_i$  and we set  $w_i = w'_i$ . Otherwise, suppose that  $w'_i$  contains an occurrence  $y$  of  $p$  that forms a factor  $f$  of length  $\ell$  ( $f$  is the  $\ell$  last letters of  $w'_i$ ). Recall that  $A_j$  is the  $j$ -th variable of  $p$ . Let  $\ell_j = |A_j|$  in the factor  $f$ , let  $L_1 = \ell_1$ ,  $L_j = L_{j-1} + \ell_j$  for  $j \geq 2$ . Let  $L' = \{L_1, L_2, \dots, L_{k-1}\}$  be a  $(k-1)$ -set of non-zero integers. Let  $X'$  be the binary word obtained from  $A_1 \cdot A_2 \cdot \dots \cdot A_k$  (where “ $\cdot$ ” is the concatenation operator) followed by as many 0's as necessary to get length  $\lfloor \frac{\ell}{2} \rfloor$ . Note that we necessarily have  $|A_1 \cdot A_2 \cdot \dots \cdot A_k| \leq \lfloor \frac{\ell}{2} \rfloor$  since the pattern is doubled. Eventually, to get  $R_i$ , we append the factor  $01^\ell$  to  $D$ ; we add  $L'$  as the last element of  $L$ ; finally we add  $X'$  as the last element of  $X$ .

**Example:** Let us give an example with  $k = 3$ ,  $p = ACBBCBBABCAB$  and  $V = [0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0]$ . The variables of  $p$  were initially ordered as  $(A, B, C)$ . For the first 24 steps, no occurrence

of  $p$  appeared, so at each step  $i \leq 24$ , we append  $V[i]$  to  $w_{i-1}$  and we append one 0 to  $D$ . Hence, at step 24, we have:

- $w_{24} = 001001100111001101110001$
- $R_{24} = \begin{cases} D = 000000000000000000000000 = 0^{24} \\ L = [ ] \\ X = [ ] \end{cases}$

Now, at step 25, we first append  $V[25] = 1$  to  $w_{24}$  to get  $w'_{25}$ . The word  $w'_{25}$  contains an occurrence  $y = (A = 01; B = 1; C = 100)$  of  $p$  which forms a factor of length 21 (the 21 last letters of  $w'_{25}$ ). Then we set  $L' = \{2, 3\}$  and  $X' = 0111000000$  (this is  $A \cdot B \cdot C$  followed by four 0's in order to get a binary word of length  $10 = \lfloor \frac{21}{2} \rfloor$ ). Eventually, to get  $w_{25}$  and  $R_{25}$ , we erase the suffix of length 21 of  $w'_{25}$  to get  $w_{25}$ , we append the factor  $01^{21}$  to  $D$ ,  $L'$  to  $L$ , and  $X'$  to  $X$ . This gives:

- $w_{25} = 0010$
- $R_{25} = \begin{cases} D = 00000000000000000000000011111111111111111111111111111111 = 0^{25}1^{21} \\ L = [\{2, 3\}] \\ X = [0111000000] \end{cases}$

This is where our example ends.

Let  $V_i$  be the vector  $V$  restricted to its  $i$  first elements. We will show that the pair  $(R_i, w_i)$  at some step  $i$  allows to recover  $V_i$ .

**Lemma 4** *After  $i$  steps of the algorithm AVOIDPATTERN, the pair  $(R_i, w_i)$  permits to recover  $V_i$ .*

**Proof.** Before step 1, we have  $w_0 = \varepsilon$ ,  $R_0 = (\varepsilon, [ ], [ ])$ , and  $V_0 = \varepsilon$ . Let  $R_i = (D, L, X)$  be the record after step  $i$ , with  $1 \leq i \leq t$ .

- Suppose that 0 is a suffix of  $D$ . This means that at step  $i$ , no occurrence of  $p$  was found: the algorithm appended  $V[i]$  to  $w_{i-1}$  to get  $w_i$ . Therefore  $V[i]$  is the last letter of  $w_i$ , say  $x$ . Then the word  $w_{i-1}$  is obtained from  $w_i$  by erasing the last letter and the record  $R_{i-1}$  is obtained from  $R_i$  by removing the suffix 0 of  $D$ . We recover  $V_{i-1}$  from  $(R_{i-1}, w_{i-1})$  by induction hypothesis and we obtain  $V_i = V_{i-1} \cdot x$ .
- Suppose now that  $01^\ell$  is a suffix of  $D$ . This means that an occurrence  $y$  of  $p$  which forms a factor  $f$  of length  $\ell$  has been created during step  $i$ . The last element  $L'$  of  $L$  is a  $(k-1)$ -set  $L' = \{L_1, L_2, \dots, L_{k-1}\}$  and the last element  $X'$  of  $X$  is a binary word of length  $\lfloor \frac{\ell}{2} \rfloor$ . Let  $\ell_1 = L_1$  and for  $2 \leq s \leq k-1$ , let  $\ell_s = L_s - L_{s-1}$ . So, for  $1 \leq s \leq k-1$ ,  $\ell_s$  is clearly the length of the variable  $A_s$  of  $p$  in the occurrence  $y$  by construction of  $L'$ . We know the pattern  $p$ , the total length of the factor  $f$  (that is  $\ell$ ) and the lengths of the  $k-1$  first variables of  $p$  in  $f$ , so we are able to compute the length  $\ell_k$  of the last variable  $A_k$ . So we are now able to recover the occurrence  $y$  of  $p$ :

the first  $\ell_1$  letters of  $X'$  correspond to  $A_1$ , the next  $\ell_2$  letters correspond to  $A_2$  and so on ( $X'$  may contain some 0's at the end which are not relevant). It follows that the factor  $f$  is completely determined. So  $w_{i-1}$  is obtained from  $w_i \cdot f$  by removing the last letter  $x$  of  $f$ , this letter  $x$  being  $V[i]$  (the appended letter to  $w_{i-1}$  at step  $i$  to get  $w'_i$ ). The record  $R_{i-1}$  is obtained from  $R_i$  as follows: remove the suffix  $01^\ell$  from  $D$ ; remove the last element of  $L$  and the last element of  $X$ . We recover  $V_{i-1}$  from  $(R_{i-1}, w_{i-1})$  by induction hypothesis and we obtain  $V_i = V_{i-1} \cdot x$ .

□

The previous lemma proves that distinct input vectors cannot correspond to the same pair  $(R_t, w_t)$ . So we get  $|\mathcal{V}| \leq |\mathcal{O}|$ .

### 4.3 Analysis of $\mathcal{R}$

Now we compute  $|\mathcal{R}|$ . Let  $R = R_t = (D, L, X)$  be a given record produced by an execution of AVOIDPATTERN. Let  $\mathcal{D}$ ,  $\mathcal{L}$  and  $\mathcal{X}$  be the set of such binary words  $D$ , of such  $(k-1)$ -sets of non-zero integers  $L$ , and of such vectors of binary words  $X$ , respectively. We thus have  $|\mathcal{R}| \leq |\mathcal{D}| \times |\mathcal{L}| \times |\mathcal{X}|$ .

Let us give some useful information in order to get upper bounds on  $|\mathcal{D}|$ ,  $|\mathcal{X}|$ , and  $|\mathcal{L}|$ . The algorithm runs in  $t$  steps. At each step, one letter is appended to  $w$ , so  $t$  letters have been appended and therefore the number of erased letters during the execution of the algorithm is  $t - |w_t|$ . At some steps, an occurrence of  $p$  appears and forms a factor which is immediately erased. Let  $m$  be the number of erased factors during the execution of the algorithm. Let  $f_i$ ,  $1 \leq i \leq m$ , be the  $m$  erased factors. We have  $|f_i| \geq q(k)$  since each variable of  $p$  is a non-empty word and  $p$  has length at least  $q(k)$ . Moreover, we have  $\sum_{1 \leq i \leq m} |f_i| = t - |w_t| \leq t$ . Each time a factor  $f_i$  is erased, we add an element to  $L$  and  $X$ , so  $|L| = |X| = m$ .

#### 4.3.1 Analysis of $\mathcal{D}$

In the binary word  $D$ , each 0 corresponds to an appended letter during the execution of the algorithm and each 1 corresponds to an erased letter. Therefore,  $D$  has length  $2t - |w_t|$ . Observe that every prefix in  $D$  contains at least as many 0's as 1's. Indeed, since a 1 corresponds to an erased letter  $x$ , this letter  $x$  had to be added first and thus there is a 0 before that corresponds to this 1. The word  $D$  is therefore a partial Dyck word. Since any erased factor  $f_i$  has length at least  $q(k)$ , any maximal sequence of 1's (which is called a *descent* in the sequel) in  $D$  has length at least  $q(k)$ . So  $D$  is a partial Dyck words with  $t$  0's such that each descent has length at least  $q(k)$ . The following two lemmas due to Esperet and Parreau [6] give an upper bound on  $|\mathcal{D}|$ .

Let  $C_{t,r,d}$  (resp.  $C_{t,d}$ ) be the number of partial Dyck words with  $t$  0's and  $t-r$  1's (resp. Dyck words of length  $2t$ ) such that all descents have length at least  $d$ . Hence, we have  $|\mathcal{D}| \leq C_{t,|w_t|,q(k)}$ .

**Lemma 5** [6]  $C_{t,r,d} \leq C_{t+d-1,d}$ .

Hence, we have  $|\mathcal{D}| \leq C_{t+q(k)-1, q(k)}$ .

Let  $\phi_d(x) = 1 + \sum_{i \geq d} x^i = 1 + \frac{x^d}{1-x}$ .

**Lemma 6** [6] *Let  $d$  be an integer such that the equation  $\phi_d(x) - x\phi'_d(x) = 0$  has a solution  $\tau$  with  $0 < \tau < r$ , where  $r$  is the radius of convergence of  $\phi_d$ . Then  $\tau$  is the unique solution of the equation in the open interval  $(0, r)$ . Moreover, there exists a constant  $c_d$  such that  $C_{t,d} \leq c_d \gamma_d^t t^{-\frac{3}{2}}$  where  $\gamma_d = \phi'_d(\tau) = \frac{\phi_d(\tau)}{\tau}$ .*

The solution of the equation  $\phi_d(x) - x\phi'_d(x) = 0$  is equivalent to  $P(x) = (1-x)^2 + (1-d)x^d + (d-2)x^{d+1} = 0$ . The radius of convergence  $r$  of  $\phi_d$  is 1 and since  $P(0) = 1$  and  $P(r) = -1$ ,  $P(x) = 0$  has a solution  $\tau$  in the open interval  $(0, r)$ . By Lemma 6, this solution is unique and, for some constant  $c_d$ , we have  $C_{t+d-1,d} \leq c_d \gamma_d^{t+d-1} (t+d-1)^{-\frac{3}{2}}$  with  $\gamma_d = \phi'_d(\tau)$ . We clearly have  $C_{t+d-1,d} = o(\gamma_d^t)$ . So, we can compute  $\gamma_d$  for  $d$  fixed. We will use the following bounds:  $\gamma_{24} \leq 1.27575$ ,  $\gamma_{40} \leq 1.15685$ , and  $\gamma_{100} \leq 1.08603$ . Note that when  $d$  increases,  $\gamma_d$  decreases.

So, by Lemmas 5 and 6, when  $t$  is large enough, we have  $|\mathcal{D}| < 1.27575^t$  (resp.  $|\mathcal{D}| < 1.15685^t$ ,  $|\mathcal{D}| < 1.08603^t$ ) if the length of any descent is at least 24 (resp. 48, 100).

#### 4.3.2 Analysis of $\mathcal{X}$

Each element  $X'$  of  $X$  corresponds to an erased factor  $f_i$  and by construction  $|X'| = \lfloor \frac{|f_i|}{2} \rfloor$ . So the sum of the lengths of the elements of  $X$  is  $\lfloor \frac{|f_1|}{2} \rfloor + \lfloor \frac{|f_2|}{2} \rfloor + \dots + \lfloor \frac{|f_m|}{2} \rfloor \leq \lfloor \frac{t}{2} \rfloor$ . Thus, the vector  $X$  corresponds to a binary word of length at most  $\lfloor \frac{t}{2} \rfloor$ . Therefore  $|\mathcal{X}| \leq 2^{\lfloor \frac{t}{2} \rfloor} \leq (\sqrt{2})^t$ .

#### 4.3.3 Analysis of $\mathcal{L}$

Each element  $L' = \{L_1, L_2, \dots, L_{k-1}\}$  of  $L$  corresponds to an erased factor  $f_i$  and by construction each  $L_j \in L'$  corresponds to the sum of the lengths of the  $j$  first variables of  $p$  in  $f_i$ .

Let  $h_k(\ell)$  be the number of such  $(k-1)$ -sets  $L'$  that correspond to factors of length  $\ell$ . Recall that  $|f_i| \geq q(k)$ , so  $h_k(\ell)$  is defined for  $k \geq 4$  and  $\ell \geq q(k)$ . Each of the  $m$  elements of  $L$  corresponds to an erased factor, so  $|\mathcal{L}| \leq h_k(|f_1|) \times h_k(|f_2|) \times \dots \times h_k(|f_m|)$ . Let  $g_k(\ell) = h_k(\ell)^{\frac{1}{\ell}}$  defined for  $k \geq 4$  and  $\ell \geq q(k)$ . Then  $|\mathcal{L}| \leq g_k(|f_1|)^{|f_1|} \times g_k(|f_2|)^{|f_2|} \times \dots \times g_k(|f_m|)^{|f_m|}$ . So, if we are able to upper-bound  $g_k(\ell)$  by some constant  $c$  for all  $\ell \geq q(k)$ , then we get  $|\mathcal{L}| \leq c^{|f_1|} \times c^{|f_2|} \times \dots \times c^{|f_m|} \leq c^t$ .

Now we bound  $g_k(\ell)$  using two different methods depending on the value of  $k$  and the length  $q(k)$  of  $p$ .

##### 4.3.3.1 Bound on $g_k(\ell)$ for $k = 4$ , $\ell \geq 100$ or $k \geq 5$ , $\ell \geq 48$

For any factor  $f_i$ , we have  $|A_1 \cdot A_2 \cdot \dots \cdot A_k| \leq \lfloor |f_i|/2 \rfloor$  since  $p$  is doubled (each variable appears at least twice). For a given  $L' = \{L_1, L_2, \dots, L_{k-1}\}$



that corresponds to some factor  $f_i$ , we have  $L_{k-1} = |A_1 \cdot A_2 \cdot \dots \cdot A_{k-1}| \leq \lfloor |f_i|/2 \rfloor$ . Therefore,  $L'$  is a  $(k-1)$ -set of distinct non-zero integers at most  $\lfloor |f_i|/2 \rfloor$ , i.e.  $k-1$  integers chosen among integers between 1 and  $\lfloor |f_i|/2 \rfloor$ ; thus  $h_k(\ell) \leq \binom{\lfloor \ell/2 \rfloor}{k-1}$  and so  $g_k(\ell) \leq \left( \frac{\lfloor \ell/2 \rfloor}{k-1} \right)^{\frac{1}{k}}$ . We can upper-bound  $g_k(\ell)$  by  $\overline{g}_k(\ell) = \left( \frac{\left( \frac{\lfloor \ell}{2} \rfloor \right)^{k-1}}{(k-1)!} \right)^{\frac{1}{k}}$  for  $\ell \geq q(k)$ . The function  $\overline{g}_k(\ell)$  is decreasing for  $\ell \geq q(k)$ ; so  $g_k(\ell) \leq \overline{g}_k(\ell) \leq \overline{g}_k(q(k)) = \left( \frac{\left( \frac{\lfloor q(k)}{2} \rfloor \right)^{k-1}}{(k-1)!} \right)^{\frac{1}{q(k)}}$  for all  $\ell \geq q(k)$ . Moreover, we can see that  $\overline{g}_k(q(k)) \leq \overline{g}_5(48)$  for all  $k \geq 5$ .

Computing this upper bound, we get  $g_k(\ell) \leq \overline{g}_k(q(k)) \leq \overline{g}_5(48) < 1.21973$  for all  $k \geq 5$  and  $\ell \geq 48$  and  $g_4(\ell) \leq \overline{g}_4(100) < 1.10456$  for all  $\ell \geq 100$ .

#### 4.3.3.2 Bound on $g_4(\ell)$ for $24 \leq \ell \leq 99$

The second method to bound the size of  $g_4(\ell)$  is based on ordinary generating functions (OGF). Here,  $k = 4$ , so let  $A_1, A_2, A_3, A_4$  be the four variables of  $p$  and let  $a_i$  be the number of apparitions of  $A_i$  in  $p$ . Therefore,  $a_1 + a_2 + a_3 + a_4 = |p|$ . Recall that each variable appears at least twice in  $p$  since  $p$  is doubled, so  $a_i \geq 2$ . Moreover, a factor of length  $\ell$ , with  $24 \leq \ell \leq 99$ , is necessarily an occurrence of a pattern of length between 24 and 99. So we just have to consider patterns  $p$  with  $24 \leq |p| \leq 99$ .

Given  $L' = \{L_1, L_2, L_3\}$  an element of  $L$  that corresponds to some factor  $f_i$ , we can compute the lengths  $\ell_i$  of each variable  $A_i$  in  $f_i$  ( $\ell_1 = L_1$ ,  $\ell_i = L_i - L_{i-1}$  for  $i \in \{2, 3\}$  and  $\ell_4 = \frac{|f_i| - (a_1\ell_1 + a_2\ell_2 + a_3\ell_3)}{a_4}$ ). Recall that  $\ell_i \geq 1$  since each variable of  $p$  is a non-empty word. Let  $\mathcal{A}_{|p|} = \sum_{i \geq |p|} b_i x^i$  be the OGF of such sets  $L'$ , i.e.  $b_i$  is the number of 3-sets  $\{L_1, L_2, L_3\}$  that corresponds to a factor of length  $i$  formed by an occurrence of a pattern of length  $|p|$  (that is  $b_i$  is the number of 4-tuples  $(\ell_1, \ell_2, \ell_3, \ell_4)$  with  $\ell_i \geq 1$  such that  $a_1 \times \ell_1 + a_2 \times \ell_2 + a_3 \times \ell_3 + a_4 \times \ell_4 = i$ ). So by definition of  $h_4$ , we have  $h_4(\ell) = b_\ell$ .

This kind of OGF has been studied and is similar to the well-known problem of counting the number of ways you can change a dollar [10]: you have only five types of coins (pennies, nickels, dimes, quarters, and half dollars) and you want to count the number of ways you can change any amount of cents. So, let  $\mathcal{C} = \sum_{i \geq 1} c_i x^i$  be the OGF of the problem and thus any  $c_i$  is the number of ways you can change  $i$  cents. Then, for example,  $c_{100}$  corresponds to the number of ways you can change a dollar. Here,  $\mathcal{C} = \frac{1}{1-x} \times \frac{1}{1-x^5} \times \frac{1}{1-x^{10}} \times \frac{1}{1-x^{25}} \times \frac{1}{1-x^{50}}$ .

In our case, we have four coins, each of them has value  $a_i$  (so we can have different types of coins with the same value) and each type of coins appears at least once (since  $\ell_i \geq 1$ ). Thus we get  $\mathcal{A}_{|p|} = \sum_{i \geq |p|} b_i x^i = \frac{x^{a_1}}{1-x^{a_1}} \times \frac{x^{a_2}}{1-x^{a_2}} \times \frac{x^{a_3}}{1-x^{a_3}} \times \frac{x^{a_4}}{1-x^{a_4}}$ . We use Maple for our computation. For each  $24 \leq |p| \leq 99$ , for each 4-tuple  $(a_1, a_2, a_3, a_4)$  such that  $\sum a_i = |p|$ , we consider the associated OGF  $\mathcal{A}_{|p|}$  and we compute, using Maple, the truncated series expansion up to the order 100, that gives  $\mathcal{A}_{|p|} = b_{24}x^{24} + b_{25}x^{25} + \dots + b_{99}x^{99} + O(x^{100})$

with explicit values for the coefficients  $b_i$ . So, for any  $24 \leq \ell \leq 99$ ,  $g_4(\ell)$  is upper-bounded by the maximum of  $(b_\ell)^{\frac{1}{\ell}}$  taken over all  $\mathcal{A}_{|p|}$ . Maple gives that  $(b_\ell)^{\frac{1}{\ell}}$  is maximal for  $|p| = 24$ ,  $(a_1, a_2, a_3, a_4) = (2, 2, 2, 18)$ , and  $\ell = 46$ : in this case,  $b_{46} = 84$  (i.e. there exist 84 distinct 3-sets  $L'$  that correspond to some factor of length 46 formed by an occurrence of a pattern of length 24). So,  $g_4(\ell) \leq 84^{\frac{1}{46}} < 1.10112$  for all  $24 \leq \ell \leq 99$ ,  $k = 4$  and  $24 \leq |p| \leq 99$ .

#### 4.3.3.3 Bound $g(\ell)$ for all $k \geq 4$

We can deduce from Paragraphs 4.3.3.1 and 4.3.3.2 the following.

If  $k = 4$ , then  $g_4(\ell) < 1.10112$  for  $24 \leq \ell \leq 99$  and  $g_4(\ell) < 1.10456$  for  $\ell \geq 100$ . So for  $k = 4$ , we have  $|\mathcal{L}| < (1.10456)^t$ .

If  $k \geq 5$ , then  $g_k(\ell) < 1.21973$  for  $\ell \geq q(k)$ . So for  $k \geq 5$ , we have  $|\mathcal{L}| < (1.21973)^t$ .

#### 4.4 Conclusion

Aggregating the above analysis, we get the following. For  $k \geq 5$ , we have  $q(k) \geq 48$ : then  $|\mathcal{R}| \leq |\mathcal{D}| \times |\mathcal{L}| \times |\mathcal{X}| \leq (1.15685 \times 1.21973 \times \sqrt{2})^t = o(2^t)$ . For  $k = 4$ , we have  $q(k) \geq 24$ : then  $|\mathcal{R}| \leq |\mathcal{D}| \times |\mathcal{L}| \times |\mathcal{X}| \leq (1.27575 \times 1.10456 \times \sqrt{2})^t = o(2^t)$ .

Thus for all  $k \geq 4$ ,  $|\mathcal{R}| = o(2^t)$  and so we obtained the desired contradiction:

$$2^t = |\mathcal{V}| \leq |\mathcal{O}| \leq 2^n \times |\mathcal{R}| = 2^n \times o(2^t) = o(2^t).$$

## 5 Conclusion

In our results, we heavily use the fact that the patterns are doubled. The fact that the patterns are long is convenient for our proofs but does not seem so important. So we ask whether every doubled pattern is 3-avoidable. By the remarks in Section 1 and by Lemma 3, the only remaining cases are doubled patterns with 4 and 5 variables. Also, does there exist a finite  $k$  such that every doubled pattern with at least  $k$  variables is 2-avoidable? We know that such a  $k$  is at least 5 since ABCCBADD is not 2-avoidable.

## References

- [1] D.R. Bean, A. Ehrenfeucht, and G.F. McNulty, Avoidable Patterns in Strings of Symbols, *Pacific J. of Math.* **85** (1979) 261–294.
- [2] J. Bell, T. L. Goh. Exponential lower bounds for the number of words of uniform length avoiding a pattern, *Inform. and Comput.* **205** (2007), 1295–1306.
- [3] J. Berstel. Axel Thue’s work on repetitions in words. Invited Lecture at the 4th Conference on Formal Power Series and Algebraic Combinatorics, Montreal, 1992, June 1992. Available at <http://www-igm.univ-mlv.fr/~berstel/index.html>.

- [4] J. Cassaigne. Motifs évitables et régularité dans les mots, Thèse de Doctorat, Université Paris VI, Juillet 1994.
- [5] V. Dujmović, G. Joret, J. Kozik, and D. R. Wood. *Nonrepetitive Colouring via Entropy*. Manuscript, 2012 (arXiv 1112.5524).
- [6] L. Esperet and A. Parreau. *Acyclic edge-coloring using entropy compression*. Manuscript, 2012 (arXiv 1206.1535).
- [7] M. Lothaire. *Algebraic Combinatorics on Words*. Cambridge Univ. Press, 2002.
- [8] R. A. Moser, G. Tardos. A constructive proof of the general lovasz local lemma. *J. ACM*, **57(2)** (2010), p. 11:1-11:15.
- [9] P. Ochem. A generator of morphisms for infinite words. *RAIRO: Theoret. Informatics Appl.* **40** (2006) 427–441.
- [10] G. Pólya, R. E. Tarjan, D. R. Woods. Notes on Introductory Combinatorics. *Progress in Computer Science*, Birkhäuser (1983).
- [11] N. Rampersad. Further applications of a power series method for pattern avoidance. *Electron. J. Combinatorics*. **18(1)** (2011), #P134.
- [12] P. Roth. Every binary pattern of length six is avoidable on the two-letter alphabet. *Acta Inform.* **29** (1992), 95–107.
- [13] A. Thue. Über unendliche Zeichenreihen. *Norske Vid. Selsk. Skr. I. Mat. Nat. Kl. Christiania*, 7:1–22, 1906.
- [14] A. Thue. Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen. *Norske Vid. Selsk. Skr. I. Mat. Nat. Kl. Christiania*, 10:1–67, 1912.
- [15] A.I. Zimin. Blocking sets of terms. *Math. USSR Sbornik* **47(2)** (1984) 353–364. English translation.